PORTING CONSIDERATIONS FOR AUTOMOTIVE MCUS

1. Introduction

The Silicon Labs MCU portfolio includes three product families that are intended for use in automotive applications. The three families are the C8051F50x/F51x, C8051F52x/F53x, and the C8051F58x/F59x. This application note highlights the differences among these product families and covers important points to consider when switching a design from one automotive MCU family to another. The C8051F50x/F51x and the C8051F58x/F59x devices were designed to be code-compatible and pin-compatible and thus require very minor changes when porting firmware and hardware between MCUs in these families.

2. Common Features

Some digital and analog peripherals are common to all automotive MCU product families. If SFR paging is accounted for, firmware written for these peripherals will work directly on any of the MCUs. See “3.2.1. SFR Paging” for more details on SFR paging.

The list of peripherals available in each of the three families is:

- LIN
- SPI
- Timers 0/1/2
- ADC and Temperature Sensor
- Comparator 0

Note that while these peripherals are common to all families, they might not be available in each part number of a product family. Refer to the Ordering Information sections of the applicable datasheets to determine the specific part number that includes the peripherals necessary for the system.

As an example, the C8051F502-GM is pin- and software-compatible with the C8051F587-GM. However, the C8051F587-GM does not include a LIN peripheral. If this peripheral is necessary for the system, the appropriate upgrade choice is the C8051F582-GM.

3. Distinguishing Factors

Table 1 lists the primary differences between the automotive MCU families. Some peripherals and capabilities are unique to certain product families. When moving a design from one MCU family to another, ensure that the new MCU family includes the necessary features. Also, note that the features listed in the table might not be available in all products in the product family. See the applicable datasheet to determine the part number that includes features necessary for the design.

Table 1. Features Differences between Automotive MCU Families

Feature	C8051F50x/F51x	C8051F58x/F59x	C8051F52x/53x
Core			
System Clock Max	50	50	25
Internal Oscillator Calibration Frequency	24 MHz	24 MHz	24.5 MHz
Program Memory	64K	128K	8K
XRAM	4K	8K	N/A
SFR Paging	✓	✓	—
Analog			
ADC Channels	32	32	16
Comparators	2	3	1
Digital			
Port Pins	40	40	16
UART (Enhanced)	✓	✓	—
UART (N/2)	—	✓	✓
CAN 2.0	✓	✓	—
External Memory Interface	✓	✓	—
Timers	4	6	3
PCA Channels	6	12	3
SMBus/I^{2}C	✓	✓	—
Pinout and Packages			
QFP48 / QFN48	✓	✓	
QFN40	✓	✓	
QFP32 / QFN32	✓	✓	
QFN-20			✓
TSSOP-20			✓
DFN-10			✓
Dedicated VIO pin	✓	✓	—

3.1. Porting Considerations between C8051F50x/F51x and C8051F58x/F59x

The ‘F58x/F59x product family was designed to be pin- and software-compatible with the 'F50x/F51x family. The 'F58x/F59x family includes all of the peripherals of the 'F50x/F51x and adds extra code space, RAM, and peripherals making it the perfect choice for 'F50x/F51x applications that require an upgrade. Corresponding package options in each product family are 100% pin-compatible and so no PCB redesign is necessary when switching between these product families.

Table 2. Pin-Compatible Automotive MCUs

Package	C8051F50x/F51x	C8051F58x/F59x
QFP48 (all are pin-compatible)	C8051F500-IQ C8051F501-IQ C8051F504-IQ C8051F505-IQ	C8051F580-IQ C8051F581-IQ C8051F584-IQ C8051F585-IQ
QFN48 (all are pin-compatible)	C8051F500-IM C8051F501-IM C8051F504-IM C8051F505-IM	C8051F580-IM C8051F581-IM C8051F584-IM C8051F585-IM
QFN40 (all are pin-compatible)	C8051F508-IM C8051F509-IM C8051F510-IM C8051F511-IM	C8051F588-IM C8051F589-IM C8051F580-IM C8051F581-IM
QFP32 (all are pin-compatible)	C8051F502-IQ C8051F503-IQ C8051F506-IQ C8051F507-IQ	C8051F582-IQ C8051F583-IQ C8051F586-IQ C8051F587-IQ
QFN32 (all are pin-compatible)	C8051F502-IM C8051F503-IM C8051F506-IM C8051F507-IM	C8051F582-IM C8051F583-IM C8051F586-IM C8051F587-IM

The addition of features to the 'F58x/F9x devices created minor differences between some common components of both families. When porting code between these families, it will be necessary to make minor firmware changes if these components are used. These differences and the required changes are detailed in Sections 3.1.1 through 3.1.4. If these changes are accounted for, any firmware written for one of the two product families can run directly on an MCU from the other product family.

3.1.1. Code Banking

The 8051 core natively supports 16-bit addressing, which allows the core to access up to 64K of Flash. In order to support Flash sizes larger than 64K, the 'F58x/F59x family implements a code banking scheme. Code banking is described in more detail in the Memory Organization section in the C8051F58x-F59x datasheet and also in application note “AN130: Code Banking Using the Keil 8051 Tools,” both of which are available at www.silabs.com.

Firmware written for the 64K/32K Flash C8051F50x/F51x devices can run on 'F58x/F59x devices without making any changes for code banking. It is only when the code size exceeds 64K that the firmware needs to account for code banking.

3.1.2. Lock Byte

On every Silicon Labs MCU, the lock byte is located in the last byte of user-accessible Flash. The 'F50x/F51x devices have 64K or 32K of Flash, and the 'F58x/F59x devices have 128K or 96K of Flash. When porting firmware from one family to another, the location of the lock byte must be moved if it is used.

Table 3. Location of Lock Byte

Device	Lock Byte Address
C8051F50x/F51x – 64K Flash	0xFBFF
C8051F50x/F51x – 32K Flash	0x7FFF
C8051F58x/F59x – 128K Flash	0x1FBFF
C8051F58x/F59x – 96K Flash	0x17FFF

3.1.3. PCA Clock Sources

Both the 'F50x/F51x and 'F58x/F59x families include a Programmable Counter Array (PCA0) peripheral with six channels. The 'F58x/F59x families add two extra clocking sources for this module: Timer 4 and Timer 5. PCA0 firmware written for the 'F50x/F51x devices will run on the 'F58x/F59x devices without any changes. PCA0 firmware written for the 'F58x/F59x devices will run on the 'F50x/F51x devices without any changes as long as Timer 4 or Timer 5 are not used.

3.1.4. C2 Programming Interface

The Flash on the MCU is typically programmed by the Silicon Labs IDE during prototyping or by a production programmer during large-scale programming.

The C2 Flash programming specification is also available for users who need to create their own Flash programming tools. See application note “AN127: FLASH Programming via the C2 Interface” for more details. Users creating their own tools for the 'F58x/F9x devices must make a small change in C2 Flash programming procedure due to the larger Flash. Before accessing Flash, the C2 register FPSEL must be set properly to select the proper bank. See the C2 Section in the C8051F58x-F59x datasheet for more details.

Programming the Flash on the 'F50x/F51x can follow the standard procedure provided in application note AN127.

3.2. Porting considerations between C8051F52x/F53x and all other automotive MCUs (C8051F50x/F51x and C8051F58x/F59x)

The differences between the C8051F52x/F53x devices and the other automotive MCU devices are more significant and require careful attention when porting firmware to and from this family, even between common peripherals. Also, the devices in the 'F52x/F53x family are much smaller than the other automotive MCUs and so there are no pin-compatible automotive devices outside of the family.

The following sections indicate the differences between the common features and peripherals that are available on the 'F52x/F53x devices and the other automotive MCUs.

3.2.1. SFR Paging

The automotive MCUs other than the 'F52x/F53x devices implement a paged SFR scheme which greatly expands the number of available SFR addresses. This SFR address expansion provides support for more peripherals and gives the programmer added flexibility. For example, PCA1CN, TMR4CN, and SCON1 in the 'F58x/F59x devices occupy bit-addressable SFR locations.

To correctly read or write to SFRs in an SFR paged device, the SFRPAGE register must be set to the correct SFR page. The SFRPAGE register itself is accessible from all SFR pages. For example, to access the crossbar initialization register XBR2, SFRPAGE must be set to 0x0F.

```
SFRPAGE = CONFIG_PAGE;                          // Switch SFR page to 0x0F
XBR2 = 0x40;                                    // Enable crossbar
```

CONFIG_PAGE is defined as 0x0F in the C8051F580.h header file. It is recommended to use the defined constants for SFRPAGE to enhance code readability and to reduce the porting effort for future platforms.

When porting code from a 'F52x/F53x device to another automotive MCU, modify the firmware to set the SFRPAGE before any SFR accesses. When porting code from the other automotive MCUs to an 'F52x/F3x device, remove all writes to SFRPAGE.

3.2.2. Oscillator and Clocking Options

On the 'F52x/F53x devices, the internal oscillator is calibrated to 24.5 MHz. On the other automotive devices, the internal oscillator is calibrated to 24 MHz. When porting code from one device to another, any firmware or peripheral that uses the internal oscillator speed to calculate timing should take into account the calibration difference.

For example, the equation to the LIN baud rate is:

$$\text{baud rate} = (\text{Systemclock})/[2^{(\text{prescaler}+1)} * \text{divider} * (\text{multiplier}+1)]$$

The firmware that initializes the LIN peripheral would need to change values written to the LIN0DIV and LIN0MUL registers.

3.2.3. UART0

The UART0 peripheral available on the 'F52x/F53x devices is different than the UART0 available on the other automotive MCUs. UART0 on the 'F52x/F53x devices is an enhanced version of the standard 8051 UART that supports a wide range of clock sources to generate standard baud rates. UART0 on the 'F50x/F51x and 'F58x/F59x is an even further enhanced UART that includes its own baud rate generator.

UART0 needs to be initialized differently on the different devices, but most firmware that writes and reads from the UART can remain the same.

The UART0 peripheral on the 'F52x/F53x devices is the same as UART1 on the 'F58x/59x devices. If the minor differences in SFR names and SFR paging are accounted for, firmware written for one of these peripherals can run directly on the other. Firmware examples for all UART peripherals are available as part of the Silicon Labs IDE installation.

3.2.4. Programmable Counter Array (PCA0)

The PCA0 peripheral on the 'F52x/F53x devices includes three channels while the other automotive MCUs include six PCA0 channels. The watchdog timer shares its functionality with the last PCA channel in PCA0. Any system that requires the use of the watchdog timer and is ported to or from a 'F52x/F53x device will need to modify the watchdog initialization.

3.2.5. Power Supply Pins

The C8051F52x/F53x devices use three pins for the power supply:

VREGIN – On-chip voltage regulator input

VDD – Digital and analog supply voltage; output of the voltage regulator

GND – Digital and analog ground

The C8051F50x/F51x and C8051F58x/F59x devices use six pins for the power supply:

VREGIN – On-chip voltage regulator input

VDD – Digital supply voltage; output of the voltage regulator

GND – Digital ground

VDDA – Analog supply voltage

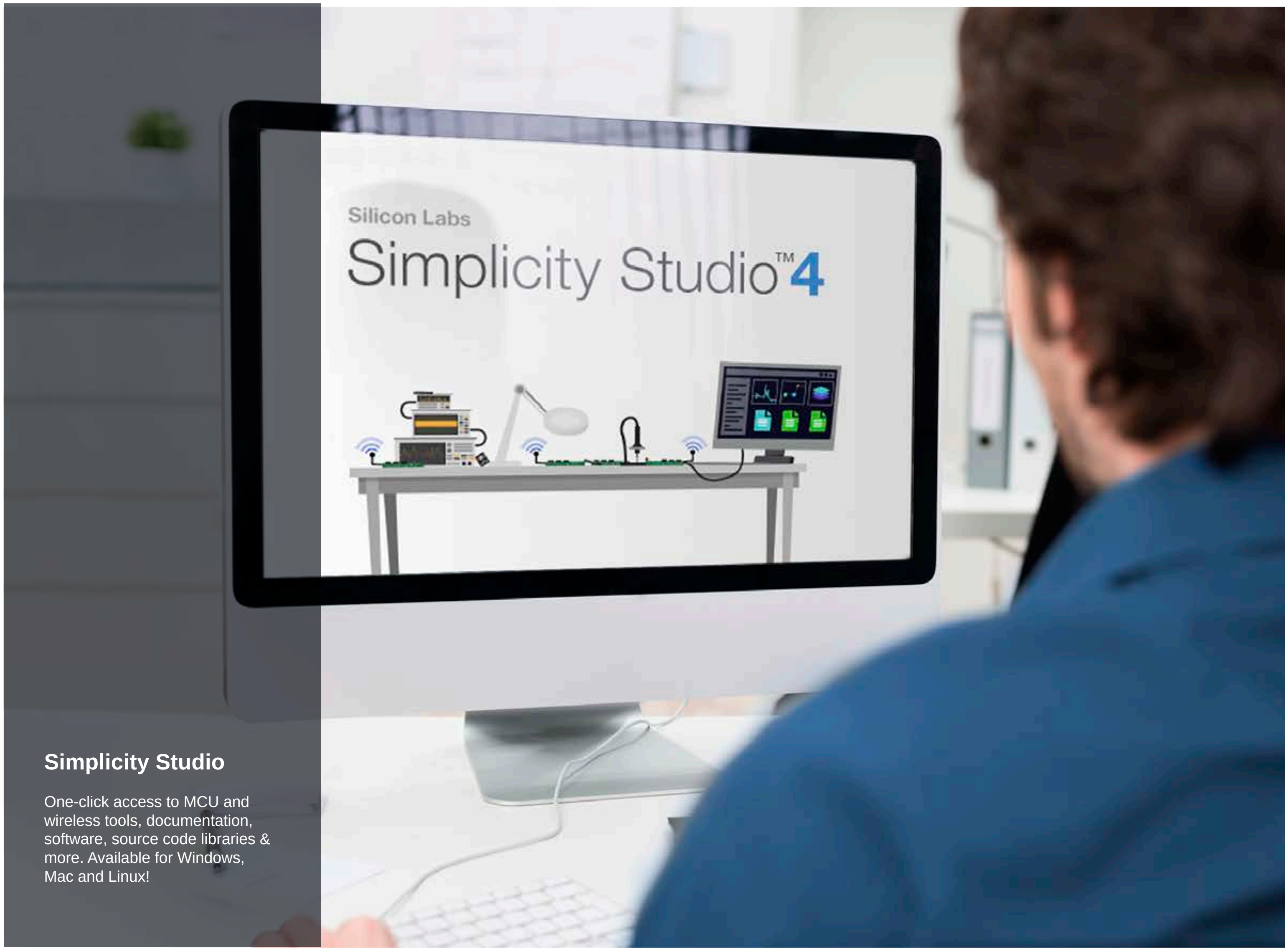
GNDA – Analog ground

VIO – Port I/O supply voltage

The 'F52x/53x devices use the supply voltage from the VDD/GND pin to power the digital core as well as the analog peripherals. The larger automotive MCUs use pins VDDA and GNDA for the analog supply and VDD and GND for the digital supply. Using these pins allow the supplies to be separately decoupled for better analog performance. VDDA and GNDA must be connected to the same potential as VDD and GND. They can be connected directly pin to pin (VDDA to VDD and GNDA to GND), or they can be isolated and connected to the same voltage.

The 'F52x/F53x devices use the voltage provided on the pin VREGIN as the power supply for the port I/O. Port pins configured for push-pull mode drive logic high signals at the VREGIN voltage, and when weak pull-ups are enabled, the pins are weakly pulled-up to this voltage. The other automotive MCUs use a VIO pin to provide the voltage for the port I/O. This added flexibility allows the device to be powered by one source through VREGIN and to interface to logic that operates at the VIO voltage.

The datasheet recommends that the C2CK/RST pin be pulled-up for noise immunity. On the 'F52x/F53dx devices, C2CK/RST should be pulled-up to VREGIN using a 4.7K Ω resistor. For devices with a VIO pin, C2CK/RST should be pulled up to VIO using a 4.7K Ω resistor.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com